

# PODSTAWY PROGRAMOWANIA W PYTHON

Dzień 6



# AGENDA

## DAY 6

- słownik dict
- praca z plikami, csv, pickle
- importowanie i używanie modułów

# | 1. Dict

# dict

## słownik {}

Zawiera pary klucz-wartość

**klucz** – musi być typem niezmiennym (string, tuple, liczba), musi być unikalny (tylko jeden w słowniku)

**wartość** – mogą być powtórzone

**Odwołujemy się poprzez klucz** a nie indeks!!!

```
x = {"nazwisko": "kowalski", "pesel": 88120134567}
```

```
x['pesel']
```

# | dict słownik {}

```
osoby = {"studenci":["Ala", "Jan", "Ania"], "wykladowcy":["doktor", "profesor"]}
```

```
print(osoby["studenci"][1])
```

```
osoby["wykladowcy"].append("magister")
```

```
osoby["administracja"] = ["pani Basia z dziekanatu"]
```

```
osoby.update({"ochrona":"Impel"})
```

```
print(osoby.keys)
```

```
print(osoby.values)
```

```
for key, item in osoby.items():
```

```
    print(key, item)
```

## 2. Praca z plikami

# pliki tekstowe

otwieramy plik

```
plik = open("ścieżka_do_pliku", tryb)
```

tryby:

**r** - tylko do odczytu

**w** - zapisywanie pliku (stary plik o tej samej nazwie będzie usunięty)

**r+** - do odczytu i zapisu

**a** - dopisywanie do pliku (dane są dopisane do końca istniejącego pliku)

# pliki tekstowe

**plik.read()** – odczytanie całego pliku, zwracany jest string zawierający cały tekst pliku (włącznie ze znakami \n) – opc. argument – **int** określająca ilość bajtów do wczytania

**plik.readline()** – odczytanie jednej linii z pliku, zwracany jest string z liniijką testu, włącznie ze znakiem \n

**plik.readlines()** – odczytuje cały tekst – zwraca listę stringów - linijek

```
for line in plik:  
    print(line, end='')
```



## with

pliki należy zamykać po użyciu:

```
plik = open(„plik.txt”)  
    # kod  
plik.close()
```

otwarcie pliku za pomocą with pozwala na automatyczne zamykanie pliku przez Pythona

```
with open(„plik.txt”) as plik:  
    print(plik.readline())
```

# pliki tekstowe

`plik.write(string)` – zapisuje string do pliku w obecnej pozycji kursora, zwraca liczbę zapisanych znaków – należy pamiętać o znaku `\n`

`plik.writelines(iterable)` – zapisuje elementy z kolekcji jako poszczególne linie w pliku

Plik musi być otworzony w trybie do zapisu aby móc go zmieniać!

# CSV

Pliki CSV – comma separated values – dane oddzielane przecinkami

Imie,Nazwisko,Adres,Telefon

Joanna,Kowalska,Gdansk Przytulna,64 654-65-45

Adam,Nowak,Gdynia Swietojanska,0700325487

Do obsługi plików CSV można użyć biblioteki csv

# pickle

pickle to moduł służący do zapisywania obiektów do plików.

Zapisać (i odczytać) możemy każdy obiekt Python'a (listy z danymi, słowniki, klasy, instancje klas (żyjące obiekty) itd..

## PICKLE UŻYCIE (TRYB BINARNY!)

```
import pickle

dane = ["Bartosz", "Mojo", 33]

with open("ogorek.pickle", "wb") as plik:
    pickle.dump(dane, plik)

# odczytanie
with open("ogorek.pickle", "rb") as plik:
    dane_wczytane = pickle.load(plik)

print(dane_wczytane)
```



# Thanks!!